論文

論文特集 ■ 最先端画像映像生成処理技法

# 空間分割モデルから境界表現への変換手法

正会員 小堀研一[†], 西尾孝治[††], 久津輪敏郎[†]

# Conversion of Spatial Partitioning Model into Boundary Representations

Ken-ichi Kobori[†], Koji Nishio[††] and Toshiro Kutsuwa[†]

**Abstract** In general, the Boundary Representations (B-Reps) model is used to create three dimensional models. In recent years, the spatial partitioning model has entered the spotlight as computer technology makes rapid progress. In particular, the Octree data structure has the advantage of having limited data space. It is necessary to convert Octree data into B-Reps to be able to use the data in conventional CAD systems. At present, the Marching Cubes method is a popular approach for such conversion. However, this method has the disadvantage that the number of generated polygons is increased. In this paper, we propose a new method to convert the Octree data structure into B-Reps. This method reduces the number of generated polygons and calculation time and provides high conversion accuracy. Several experimental results show that this method is effective in converting Octree data structure into B-Reps.

## 1. まえがき

従来，3次元CAD・CG分野では形状モデルとして境界表現が広く用いられてきた．境界表現はデータの構成要素が形状の頂点，稜線および面であり，データ効率が高い．また，表示アルゴリズムもほぼ確立されており，レンダリングの際に計算機にかかる計算負荷が小さいといった利点がある．しかし，ユーザは形状操作において形状の頂点，稜線および面を意識しなければならない，また，立体集合演算などの形状処理のアルゴリズムが複雑になるなどの問題がある．

一方，近年CAEの可視化，医用分野などでボクセルに代表される空間分割モデルが注目されている[1)2)]．空間分割モデルはデータ構造が簡単であるため，立体集合演算などの形状操作が容易であり，境界表現のように形状の構成要素を意識する必要がないなどの利点がある．しかし，境界表現に比べてデータ効率が低い，レンダリングの負荷が大きいといった問題もある．前者のデータ効率が低いという問題に対しては，ボクセルではなくオクトツリーデータ構造を用いるこ

キーワード：形状モデル，オクトツリー，マーチン・キューブス法，CAD，境界表現，モデル変換
1994年9月30日，第49回（平成6年後期）情報処理学会全国大会で発表
1995年2月22日受付，1995年5月22日再受付
† 大阪工業大学 電子工学科（〒535 大阪市旭区大宮5-16-1，TEL 06-954-4307）
†† 大阪工業大学 電気工学専攻（同上）
† Department of Electoronic Engineering, Osaka Institute of Technology (5-16-1, Ohmiya, Asahi-ku, Osaka 535, Japan)
†† Department of Electrical Engineering, Osaka Institute of Technology (ditto)


テレビジョン学会誌 Vol. 49, No. 10, pp. 1230～1239 (1995)

とによりデータ効率の改善をはかる方法[1]も提案されている．

そこで，境界表現とオクトツリーの長所を合わせ持つハイブリッドモデルも提案されている[3)4)]．このようなハイブリッドモデルの実現には，境界表現-オクトツリー間の双方向変換が必要となる．これまでに境界表現からオクトツリーへの変換手法についてはいくつかの手法が報告されている[5)~8)]．

また，空間分割モデルから境界表現への変換手法については，ボクセルを対象としたマーチン・キューブス法[9)10)]がよく知られている．しかし，従来のマーチン・キューブス法では，精度を上げるためにボクセルの大きさを小さくすると変換後に生成される面数が膨大になり，データ効率やレンダリング処理に対する負荷の増加などの点で問題があった．これは，マーチン・キューブス法では，空間分割モデルが構成する形状の複雑さに関係なく一様に細かな面を生成しているためである．

また，ランレングスを3次元に拡張した3D-ランレングスを用いてボクセルを境界表現へ変換する手法[11)]も提案されているが，ボクセルを単位として境界面を構成するため生成面数が膨大になるという問題点は残る．

そこで，本論文では従来のマーチン・キューブス法に比べ，変換精度を低下させることなく生成面数を減少させた，オクトツリーから境界表現への変換手法について述べる．

## 2. オクトツリー-境界表現変換の概要

オクトツリーは8分木構造を持っているが，本論文ではルートのオクタントを分割レベル0のオクタントと呼ぶことにする．次に，レベル0のオクタントから分岐しているブランチのオクタントを分割レベル1のオクタントと呼び，以下同様に，$n$回分岐したオクタントを分割レベル$n$のオクタントと呼ぶ．また，メモリーの容量には限界があるので，これ以上分割しないという分割レベルを設定し，これを最大分割レベルと呼ぶ．また，オクトツリーの8分木構造のリーフに相当するオクタントをリーフオクタント，ブランチに相当するオクタントをブランチオクタントと呼ぶ．

本手法で変換後に得られる境界表現は，すべて三角形ポリゴンで構成する．このポリゴンには従来のマーチン・キューブス法で用いられている三角形ポリゴンを採用する．以下，この三角形ポリゴンをマーチン・キューブス・パッチと呼ぶ．

図1に本手法の流れを示す．また，以下に本手法の概要を述べる．

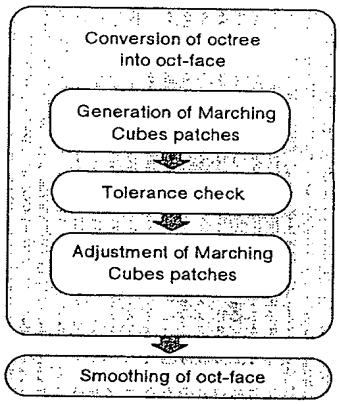



図1 処理概要
General flowchart.

まず，分割レベルがレベル0のオクタントにおいてマーチン・キューブス・パッチを生成する．

次に，生成されたマーチン・キューブス・パッチがオクトツリーが構成する形状を表現するのに妥当かどうかの判定をする．妥当であれば，生成されたマーチン・キューブス・パッチを生成面として採用する．妥当でない場合は，生成されたマーチン・キューブス・パッチを破棄し，新たに1つ高い分割レベルのオクタントにおいてマーチン・キューブス・パッチを生成し，再びオクトツリーが構成する形状に対する妥当性を調べる．

このように，分割レベルの低いオクタントから分割レベルの高いオクタントへ順次，マーチン・キューブス法によって面を生成していく．

最後に，生成したマーチン・キューブス・パッチの整合処理を行う．

変換処理はこれで完了するが，生成形状の表現精度を上げるために，本手法ではスムージング処理を加える．

従来のマーチン・キューブス法では，最大分割レベルでのみマーチン・キューブス・パッチを生成していたので必要以上に細かな面を生成していたが，本手法はオクトツリーが表現する形状の表面で曲率の低い部分では大きなマーチン・キューブス・パッチを生成し，曲率の高い部分では細かなマーチン・キューブス・パッチを生成することができる．

## 3. データ構造

オクトツリーから境界表現への変換はオクタント単位で行う．このため，各オクタントで生成された面を格納するデータ構造としてオクトフェイスデータ構造

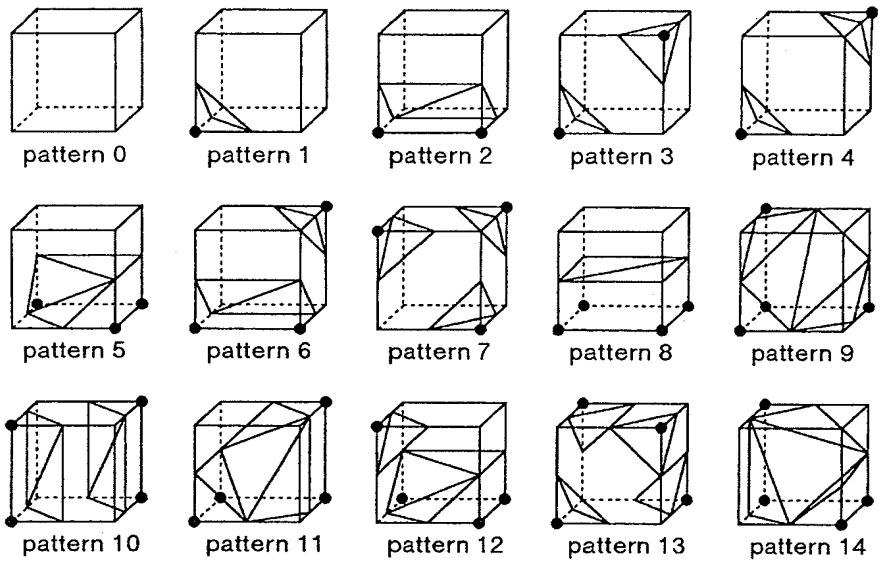



図 2　マーチン・キューブス・パターン
15 patterns of Marching Cubes patches.

を提案する．オクトフェイスデータ構造は，オクトツリーデータ構造と同様の 8 分木データ構造を持つ．

### 3.1　オクトツリーデータ構造

オクトツリーデータ構造は空間分割モデルの一種で，可変の大きさの立方体で表されるオクタントを構成単位とした集合である．

オクトツリーは形状の境界付近で分割が繰り返されるため，形状表面から離れた領域では分割が繰り返されず，3 次元空間を格子状に分割するボクセルに比べるとデータ効率の向上がはかれる．

以下，本論文では，形状内部に相当するオクタントを Black オクタント，形状外部に相当するオクタントを White オクタント，また，形状表面に干渉しているオクタントを Gray オクタントと呼ぶ．

### 3.2　マーチン・キューブス法

マーチン・キューブス法[10]は，空間分割モデルを微小な三角形ポリゴンの集合に変換する手法である．この手法を用いれば，対象とする空間分割モデルを境界表現に変換することができる．一度境界表現に変換すると，従来の境界表現に対する表示処理の手法をそのまま利用することができるため，一般に空間分割モデルの表示の前処理として用いられる．

しかし，空間分割モデルの精度を上げると生成面数が急激に増加し，データ量が膨大になる，変換処理の負荷，およびレンダリング負荷が大きくなるなどの問題がある．

次に，ボクセルデータ構造を例にとり，この処理手順について簡単に述べる．ボクセルには濃度値があるものとする．対象とするボクセルデータのうち，8 つのボクセルを 1 つの単位として，各セルの濃度値があらかじめ設定した閾値より大きいか小さいかを調べ，図 2 に示すような 15 通りのマーチン・キューブス・パターンに当てはめる．このパターンから最大 4 枚のマーチン・キューブス・パッチを生成する．生成されたマーチン・キューブス・パッチの頂点座標はボクセルの濃度値と閾値を用いて線形補間する．

このような処理を 8 つのボクセルを単位として行うことで，ボクセルデータを微小な三角形平面の集合で表される境界表現へ変換する．

### 3.3　オクトフェイスデータ構造

本論文で提案するオクトツリーから境界表現への変換には，変換後のデータ構造としてオクトツリーにマーチン・キューブス法を用いて生成した最大 4 枚のマーチン・キューブス・パッチの情報を付加したデータ構造を導入する．以下，このデータ構造をオクトフェイスデータ構造と呼ぶ．オクトフェイスデータ構造のオクタントには，次の(ⅰ)〜(iii)の 3 つの状態がある．

(ⅰ)　図 3(a)は，リーフオクタントであり，マーチン・キューブス・パッチを持っている状態である．

(ii)　図 3(b)は，リーフオクタントであり，3 次元空間でオクタントの占める領域がオクトツリーの形状に対して完全に内部，もしくは外部に相当し，マーチン・キューブス・パッチを持たない状態である．

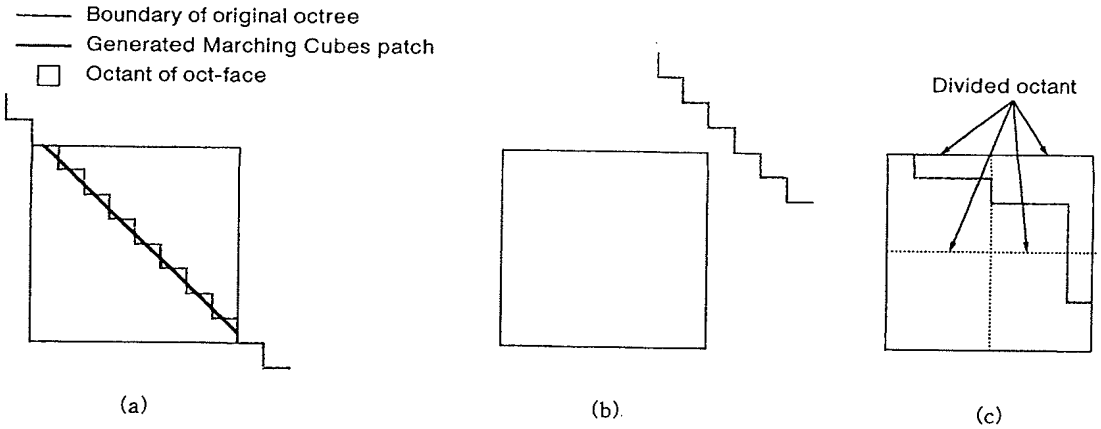



図 3　オクトフェイスのオクタントの状態
3 states of oct-face octant.

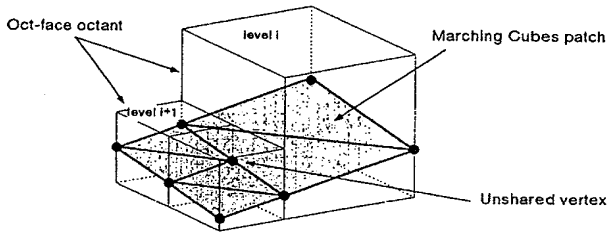



図 4　オクトフェイスにおける共有しない頂点
Unshared vertex of Marching Cubes patch in oct-face.

(iii)　図3(c)は，3次元空間でオクタントの占める領域がオクトツリーの形状表面に対して干渉しており，注目している分割レベルのマーチン・キューブス・パッチでは充分に形状を表現できない場合である．この場合は，このオクタントを $XYZ$ の各軸に対してオクタントを2等分することで8つの子オクタントを生成する．したがって，注目しているオクタントはブランチオクタントとなる．

また，本論文においてオクトフェイスデータ構造では，図4に示すように隣接オクタント間で分割レベルが異なる場合には，隣接パッチ間で必ずしも頂点を共有しないことを許す．

## 4. オクトツリーから境界表現への変換手順

最初に，オクトツリーからオクトフェイスデータ構造へ変換し，生成されたオクトフェイスデータ構造からマーチン・キューブス・パッチで表される三角形ポリゴンの情報のみを取り出せば境界表現を得ることができる．オクトフェイスデータ構造からポリゴン情報を得ることは容易なので，本章ではオクトツリーからオクトフェイスデータ構造への変換について述べる．

処理手順は以下の3つからなる．

①　オクトツリーからオクトフェイスデータ構造への変換処理

オクトツリーをもとにマーチン・キューブス法を用いて，あらかじめ設定した誤差基準の範囲内でオクトフェイスデータ構造を生成する．

②　整合処理

①で生成されたオクトフェイスデータ構造には隣接マーチン・キューブス・パッチ間に最大分割レベルのオクタントの幅に相当する隙間が生じているので，この隙間を埋める．

③　スムージング処理

一般に，マーチン・キューブス法を用いると生成形状に段階状の凹凸を生じることが知られている．この階段状の凹凸を滑らかにする．

なお，各処理はオクタント単位で行われる再帰処理である．以下に，これらの処理について詳細を述べる．

### 4.1 オクトツリーからオクトフェイスデータ構造への変換処理

オクトツリーからオクトフェイスデータ構造への変換は，オクトツリーのオクタント単位でマーチン・キューブス・パッチを生成する．このパッチがオクトツリーの形状を表現するのに妥当かどうかを判定して，分割レベルが低いオクタントから分割レベルが高いオクタントへ処理を移す．

(1)　マーチン・キューブス・パッチ生成

この処理では，オクトツリーを構成するオクタントに注目し，オクタント単位でマーチン・キューブス・パッチを生成する．図5(a)の任意の分割レベルのオクタントを例に説明する．このオクタントのリーフオ

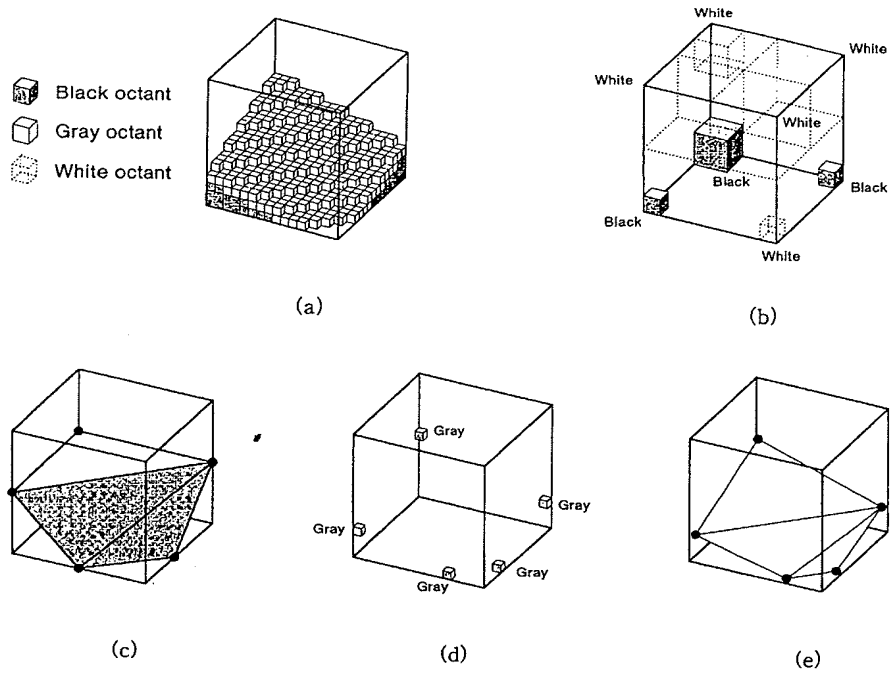



図 5　オクトツリーからオクトフェイスの生成
Generation of oct-face from original octree.

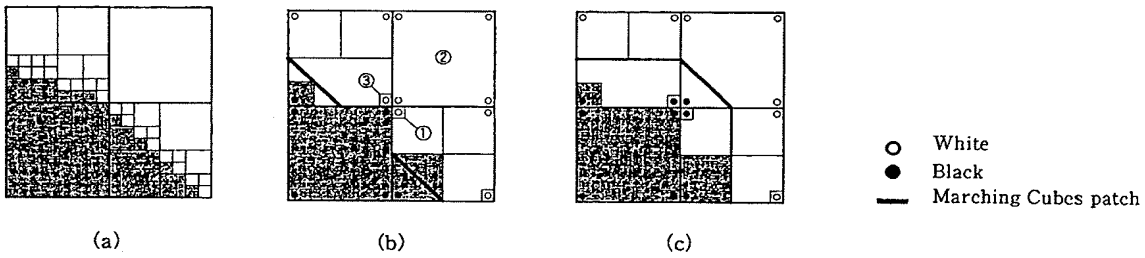



図 6　オクトフェイス生成における規則
A rule of oct-face generation.

クタントのうちで図5(b)に示すように8隅に相当する8つのオクタントのモードを調べる．

ただし，これらのオクタントのうちでモードがGrayのものについては，そのモードをBlackとみなす．

次に，マーチン・キューブス法を用いてマーチン・キューブス・パターンを生成する．従来のマーチン・キューブス法では，空間分割モデルのセルが持つ濃度値があらかじめ設定された閾値より大きいか小さいかでマーチン・キューブス・パターンを生成していたが，本手法で取り扱うオクトツリーではその濃度値を持っていない．そこで，この8隅に相当するオクタントのモードを用いてマーチン・キューブス・パターンを生成する．これにより，図5(c)に示すような，オクトツリーが表す形状に相当するマーチン・キューブス・パターンを得ることができる．また，従来のマーチン・キューブ法では，生成されるマーチン・キューブス・パッチの頂点座標を，空間分割モデルが持つ濃度値とあらかじめ与えられた閾値により線形補間を行っていたが，本手法では，図5(d)に示すように最大分割レベルのGrayオクタントの中心座標で補間する．生成されたマーチン・キューブス・パッチを図5(e)に示す．

なお，8隅のオクタントのモードを調べる際に以下のような条件を加える．

図6の2次元の例で説明する．図6(a)のようなオクタントが与えられた場合，4つのオクタントの4隅の状態は図6(b)のようになり，生成されるマーチ

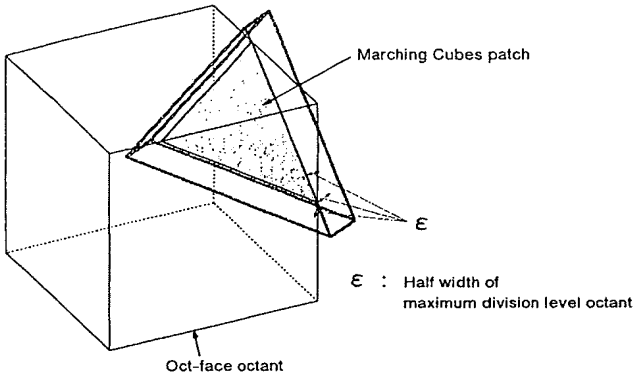



図 7 誤差判定
Tolerance check.

ン・キューブス・パターンは不連続となる．このような状態を回避するために，オクタント①に隣接する3つのオクタント②③④のうち1つでもモードがBlackである場合はオクタント①のモードはBlackとする．他のオクタントについても同様に隣接する3つのオクタントを調べ同様のルールを適用する．これにより，図6(c)に示すような連続したマーチン・キューブス・パターンを得ることができる．

(2) 誤差判定

生成されたマーチン・キューブス・パッチの形状の妥当性に対する基準として以下のような誤差判定を行う．生成されたマーチン・キューブス・パッチをもとに図7に示すように，パッチから最大分割レベルのオクタントの幅に相当する距離だけ離れた2面および，パッチを構成する稜線から最大分割レベルのオクタントの幅に相当する距離だけ離れ，かつパッチに垂直な3面で三角柱をつくる．注目しているオクタントが複数枚のマーチン・キューブス・パッチを持つ場合は，このような三角柱をパッチの枚数分つくる．注目しているオクタントのすべてのGrayの状態のリーフオクタントの中心座標がこのような三角柱のいずれかに含まれていれば，生成されたマーチン・キューブス・パッチはオクトツリーの形状を表現するのに妥当であるものとし，オクトフェイスデータ構造の生成面として採用する．もし，1つでもこれらの三角柱のいずれにも含まれないGrayの状態のリーフオクタントがあれば，このマーチン・キューブス・パッチはオクトツリーの形状を表現するのに妥当ではないものとして破棄し，オクタントを分割し，1つ高い分割レベルで再びマーチン・キューブス・パッチを生成し，同様の誤差判定を行う．また，オクタントの分割を繰り返すことで分割レベルが最大分割レベルより1つ低い分割レベルに達した場合は，誤差判定は行わずに生成されたマーチン・キューブス・パッチをオクトフェイスデータ構造の生成面として採用する．これれらの処理を再帰的に繰り返すことにより，オクトフェイスデータ構造を生成することができる．

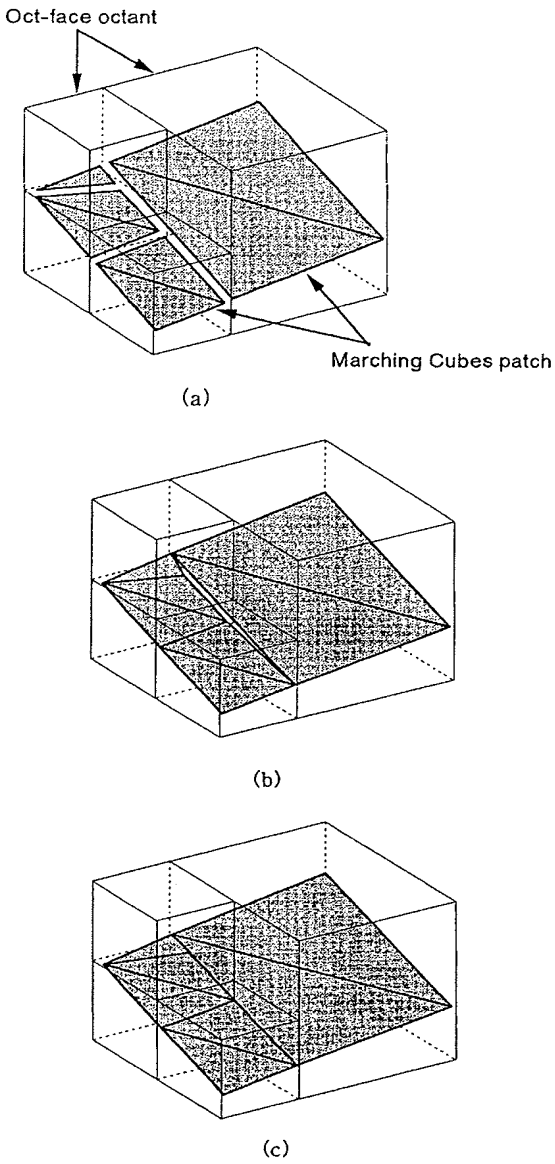



図 8 整合処理
Adjustment of Marching Cubes patches in adjacent oct-face octants.

### 4.2 整合処理

生成されたオクトフェイスデータ構造が持つパッチは，前に述べたようにその頂点座標を最大分割レベルのGray状態のオクタントの中心座標で補間している．このため，図8(a)に示すように，隣接オクタントが持つパッチとの間で最大分割レベルのオクタントの幅に相当する隙間が生じる．これでは，境界が不連

続になり境界表現としては不充分である．そこで，この隙間をなくすために，オクトフェイスデータ構造の整合処理を行う．

図8(a)のように隣接オクタントの稜線どうしが接している部分では，隣接するマーチン・キューブス・パッチの頂点の平均座標を求め，この点に各マーチン・キューブス・パッチの頂点を移動する．この操作により，図8(b)に示すように部分的にマーチン・キューブス・パッチの隙間を埋めることができる．

しかし，この処理だけでは不充分で，一般に隣接オクタント間では分割レベルが異なるために，図8(b)に示すような隙間をなくすことはできない．そこで，このような場合は，分割レベルの高いオクタントのパッチの頂点を，この頂点から分割レベルの低いオクタントのパッチの稜線上に下した垂線との交点に移動する．

この処理により，図8(c)に示すようにすべての隙間を埋ることができる．

### 4.3 スムージング処理

4.1節では，オクトフェイスデータ構造におけるマーチン・キューブス・パッチの生成方法を述べた．しかし，その方法で生成したオクトフェイスデータ構造の形状には階段状の凹凸が生じる部分がある．

4.1(1)で述べたように，マーチン・キューブス・パターンを生成するには8隅のオクタントが必要となる．この要件を満たす最も高い分割レベルは，8つの最大分割レベルの子オクタントを持つ状態，すなわち最大分割レベルより1つ低い分割レベルである．このレベルで生成したマーチン・キューブス・パッチが最も細かなものとなる．

次に，最大分割レベルより1つ低い分割レベルでマーチン・キューブス・パッチを生成する場合を例にとり，階段状の凹凸が生じる問題点を説明する．

この分割レベルで生成したマーチン・キューブス・パッチの頂点座標の補間要素となるGrayのリーフオクタントは，注目しているオクタントの子オクタントである．また，注目しているオクタントの子オクタントがとりうる座標は8通りしかない．このため，補間後のマーチン・キューブス・パッチの頂点座標も8通りとなる．したがって，このような分割レベルで生成されたマーチン・キューブス・パッチを含む部分では生成形状に階段状の凹凸が生じる．

そこで，この凹凸を滑らかにするためにスムージング処理を行う．この処理は，図9(a)に示すように処理中のオクタントに隣接するオクタントが持つパッチの法線ベクトル，およびパッチの中心座標の平均値を

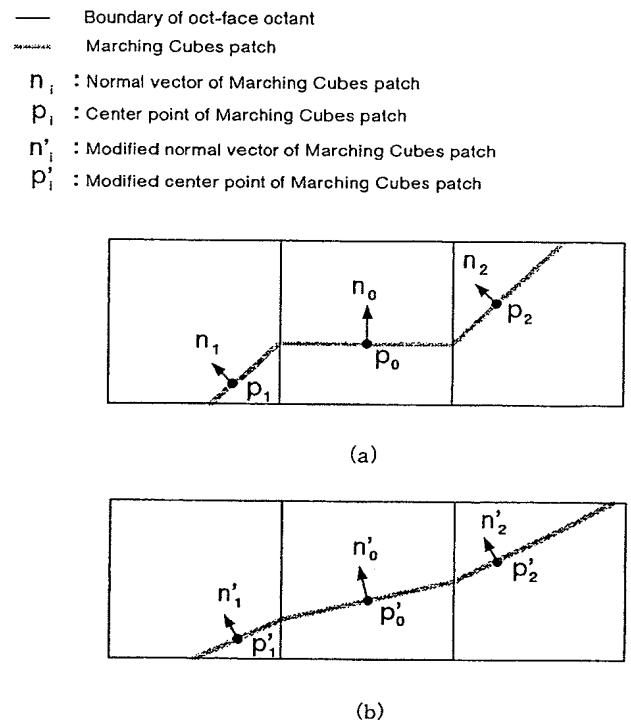
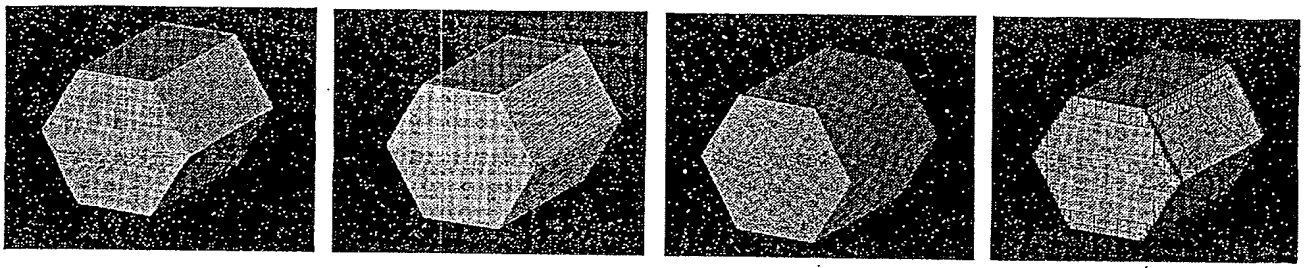



図9 スムージング処理
Smoothing of generated oct-face.

図9(b)に示すようにスムージング処理後のパッチの法線ベクトルおよび中心座標として更新する．

## 5. 実験結果

オクトフェイスデータ構造を用いた本手法の有効性を検証するために，空間分割モデルから境界表現への変換において従来法との比較実験を行った．なお，オクトツリーから境界表現への変換手法は確立されておらず，直接本手法の比較対象となる手法はないため，ボクセルから境界表現への変換手法であるマーチン・キューブス法を従来法として比較対象にした．ここで，従来法としてボクセルの代わりにオクトツリーの最大分割レベルより1つ低い分割レベルのオクタントを単位として，このオクタントから分岐している8つの最大分割レベルのオクタントにマーチン・キューブス法を適用し，マーチン・キューブス・パッチを生成した．実験は，従来法，本手法ともにレベル6からレベル8まで最大分割レベルを変化させて六角柱形状のオクトツリーデータから境界表現を生成したときの生成面数，および変換にかかる処理時間を計測した．実験にはSiliconGraphics社のIndy(約150 MIPS)を使用した．写真1(a)に変換対象としたオクトツリーで表現された形状を示す．このオクトツリーを境界表現



(a) (b) (c) (d)

写真 1 六角柱
Hexagonal prism.

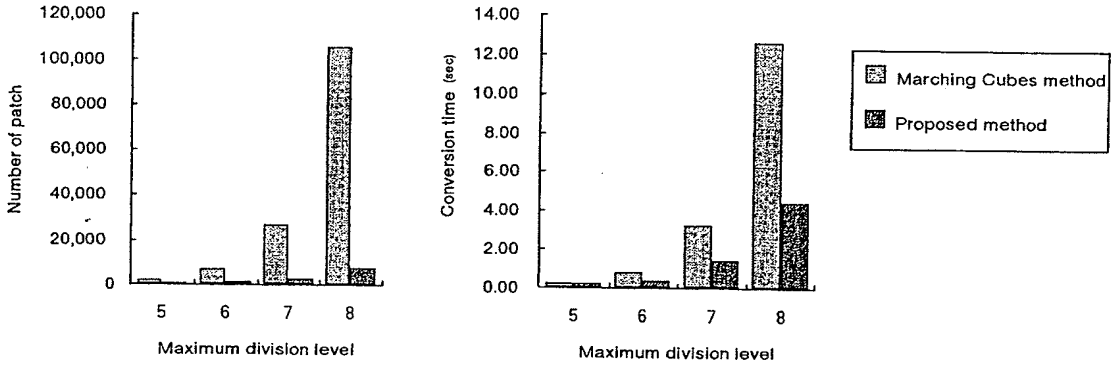


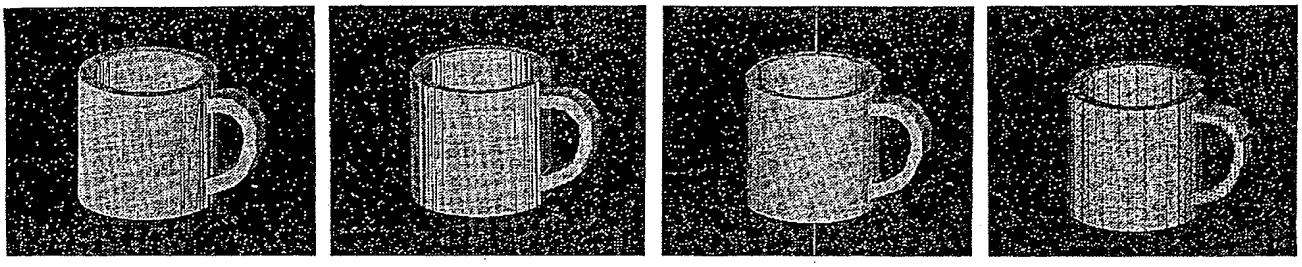

図 10 生成面数と変換時間
Statistics for conversion of the method.

表 1 変換結果
Result of the conversion.

| Model | | | Polyethylene Vessel | Cup | Camera |
|---|---|---|---|---|---|
| Number of Patches | Marching Cubes method | | 60,392 | 75,348 | 110,692 |
| | Proposed method | | 14,240 | 8,512 | 20,018 |
| Conversion Time (sec) | Marching Cubes method | | 7.23 | 10.96 | 14.22 |
| | Proposed method | Coversion | 5.11 | 4.95 | 8.77 |
| | | Smoothing | 0.90 | 0.42 | 1.40 |
| | | Total | 6.01 | 5.37 | 10.17 |

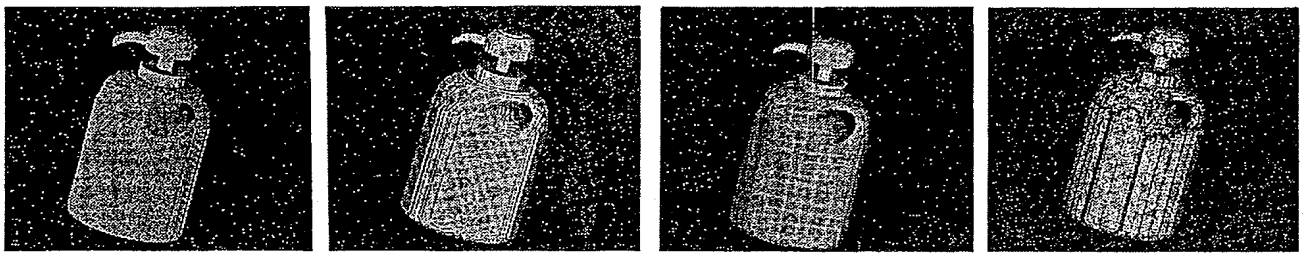
へ変換して得られたポリゴン数と，変換処理に要した時間を図 10 に示す．従来のマーチン・キューブス法によって得られた形状を写真 1 (b)に，本手法によって得られた形状を写真 1 (c)に示す．また，生成されたポリゴンを示すために，本手法により得られた形状にワイヤフレームを重ねたものを写真 1 (d)に示す．これらのモデルの最大分割レベルは，いずれも分割レベル 8 である．

さらに，いくつかの製品形状についても変換実験を行った．形状としてはマグカップ形状，ポリ容器形状，およびカメラ形状を最大分割レベル 8 のオクトツリーで作成し，従来法と本手法によるオクトツリーから境界表現への変換実験を行った．変換により生成された面数，および変換処理時間を表 1 に示す．これらのモデルのオクトツリー形状を写真 2～写真 4 の(a)に，従来法による変換結果を写真 2～4 の(b)に，本手法による変換結果を写真 2～4 の(c)に，本手法により得られた形状にワイヤフレームを重ねたものを写



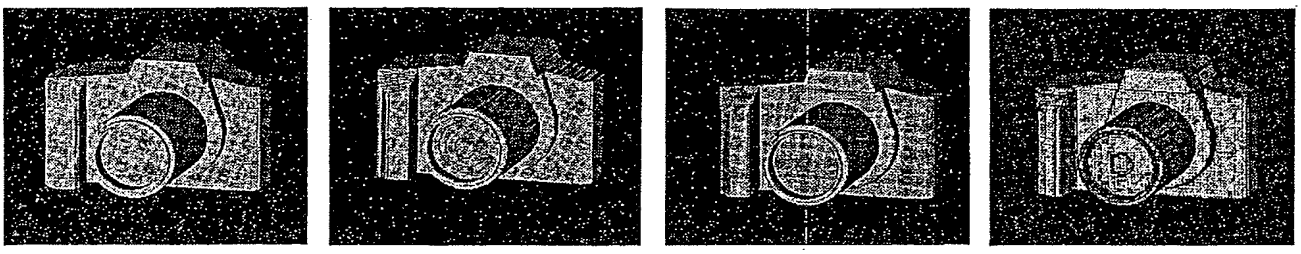
(a) (b) (c) (d)

写真 2 コップ形状
Cup model.



(a) (b) (c) (d)

写真 3 ポリ容器形状
Polyethylene vessel model.



(a) (b) (c) (d)

写真 4 カメラ形状
Camera model.

真2～4の(d)に示す．

## 6. 考　　察

まず，図10のグラフから設定したいずれの最大分割レベルにおいても，従来のマーチン・キューブス法に比べて本手法は生成面数が少なく，処理時間が短いことがわかる．また，従来ボクセルによるマーチン・キューブス法では，最大分割レベルを1レベル上げるごとに生成面数は形状の表面積に比例するため約$2^2$倍になる．これに伴い処理時間も約4倍になる．これに対し，オクトツリーによる本手法では生成面数の増加率が最も大きい場合で約4倍であり，最も小さい場合は分割が起こらず増加しないため，分割レベルを高くしても従来のマーチン・キューブス法ほど生成面数は増加しない．実験結果でも，本手法では生成面数の増加率が約2倍，処理時間の増加率が約2.3倍程度に押さえられている．以上のことから，本手法は最大分割レベルが高い，すなわち高い精度を要求するモデルになるほど有効であるといえる．

ただし，最大分割レベルが低い場合は，本手法による変換処理時間が従来法とあまり変わっていない．最大分割レベルが低い場合はオクタント数が少なく，本手法の特徴である面積の大きなパッチがあまり生成されず，比較的均一な大きさのパッチが生成される．こ

のため生成面数が従来法とあまり変わらず，誤差判定にかかる時間が処理時間の多くの部分を占めることになり，変換にかかる総処理時間は短縮されなかったものと考えられる．

意匠設計など高い精度を必要とするCAD分野で空間分割モデルを用いるためには，500 mmの形状で1 mmの精度，すなわち最低でも分割レベル9程度の最大分割レベルが必要と考えられるため，変換処理時間，生成されるパッチ数において，本手法は空間分割モデルから境界表現への変換に有効であることが明らかとなった．

次に，従来のマーチン・キューブス法と本手法によって得られた境界表現について比較してみる．本論文で扱っている形状外部と内部といった2値情報しか持たない空間分割モデルに対して従来のマーチン・キューブス法を用いると，変換後に得られるマーチン・キューブス・パッチの頂点座標を補間する要素がないため，形状表面に階段状の凹凸が生じている．

一方，本手法によって得られた形状は，同一平面とみなせる部分，もしくは曲率の低い部分では比較的大きなマーチン・キューブス・パッチが生成されており，形状表面にこのような凹凸は生じていない．また，オクトフェイスデータ構造における最大分割レベルで生成されたマーチン・キューブス・パッチは，頂点座標の補間要素がないために形状表面に凹凸が生じると考えられるが，スムージング処理によってこのような凹凸の発生を避けることができた．

## 7. むすび

本手法は，変換処理時間，生成面数および生成形状の再現性から，空間分割モデルから境界表現への変換に有効であるということが明らかとなった．また，オクトツリーの分解能が高くなるほど本手法の有効性が顕著になる．

しかし，本手法では主に形状表面の曲率の大きい部分や形状のエッジ付近でオクトフェイスデータ構造の分割が繰り返されるため，細かなマーチン・キューブス・パッチが生成されている．このような部分を認識して分割を繰り返さずに比較的大きな面を生成しエッジを表現する，もしくは，このような細かな面をまとめて大きな面で再構成して形状を表現する処理を行い，さらに生成面数の減少を図ることを考えている．また，オクトツリー-境界表現間の双方向変換が実時間で行えれば，両モデルの長所を生かしたハイブリッドモデルによる対話的なモデリングが可能となるので，最大分割レベルでの変換処理のより高速化などが今後の課題として考えられる．

## 〔参考文献〕


1) F. Foley, A. VanDam, S. Feiner and J. Hughes: "Computer Graphics Principles and Practice", pp. 553-562, Addison-Wesley Publishing Company (1990)
2) 藤代, 茅, 國井："ボクセル志向3次元データ表現とその表示技術", 情処学誌, **34**, 3, pp. 285-298 (1993)
3) 小堀："工業デザインのための新しいモデリングアプローチ", 日本機械学会第6回計算力学講演会論文集, pp. 475-476 (1993)
4) Pure Brunet, Isabel Navazo: "Solid Representation and Operation Using Extended Octrees", ACM Transactions on Graphics, **9**, 2, pp. 170-197 (1990)
5) Franklin, W. R. and Akman, V.: "Building an Octree from a Set of Parallele-pipeds", IEEE Computer Graphics and Applications, **5**, 10, pp. 58-64 (1985)
6) M. Tamminen, H. Samet: "Efficient Octree Conversion by Connectivity Labelling", Comput. Graphics, **18**, 3, pp. 43-51 (July 1984)
7) 登尾, 福田, 有本："Brepsからオクトツリーへの変換アルゴリズムとその評価", 情報学論, **28**, 10, pp. 1003-1012 (1987)
8) 石黒, 小堀, 久津輪："多面体データを空間分割モデルへ高精度変換する一手法", 1993信学秋季大, 6-325 (1993)
9) William E. Lorensen, Harvey E. Cline: "Marching Cubes a High Resolution 3D Surface Construction Algolithm", Comput. Graphics, **21**, 4 (July 1987)
10) 石井, 安田, 横井, 鳥脇："マーチングキューブ法の改良アルゴリズムについて", 情処学会のグラフィクスとCADシンポジウム, 60-9, pp. 63-70 (1992)
11) 荒川："仮想空間における立体形状モデリング", 情処学会, グラフィックスとCADシンポジウム, pp. 33-42 (Nov. 1991)



**小堀 研一**（こぼり けんいち） 昭和50年，山梨大学大学院工学研究科修了．同年，シャープ(株)に入社．以後，CAD，CGに関する研究開発に従事．平成3年，大阪工業大学電子工学科助教授となり，現在，同学科教授．CAD，CG，バーチャルリアリティに関する研究に従事．工学博士．正会員．

**西尾 孝治**（にしお こうじ） 平成6年，大阪工業大学電子工学科卒業．現在，同大学院工学研究科博士前期課程在籍中．3次元CADシステムの形状モデルに関する研究に従事．

**久津輪敏郎**（くつわ としろう） 昭和48年，府立大阪大学大学院博士課程修了．同年，大阪工業大学電子工学科講師．51年，助教授．59年，教授．計算機アーキテクチャ，故障診断，論理設計，CAD等の研究に従事．工学博士．